LISTING OF CLAIMS

1. (Canceled)

2. (Currently amended) The ~~multiple-input queuing~~ system of claim 1<u>6</u>, ~~further including~~<u>wherein said input means comprises</u>
———a first switch<u>,</u> ~~operably coupled to the allocator, that is~~ configured to route the data-~~item~~ <u>element</u> from ~~the select input streams~~<u>said one of the input data streams</u> to ~~the~~ ~~memory elements~~<u>said unused data memory location</u>.

3. (Currently amended) The ~~multiple-input queuing~~ system of claim 2, ~~further including~~<u>wherein said readout circuit comprises</u>
———a second switch<u>,</u> ~~operably coupled to the mapper, that is~~ configured to route the ~~data-item~~<u>next data element</u> from the ~~memory element~~<u>array of data memory locations</u> to the output.

4. (Currently amended) The multiple-input queuing system of claim 1<u>6</u>, wherein<u> the input means comprises an</u>
———<u>the </u>allocator ~~is further~~ configured to allocate ~~the memory element~~<u>an unused data memory location</u> based on a request from ~~the~~<u>a </u>selected input <u>data</u>-stream for an allocation.

2

5. (Currently amended) The ~~multiple-input queuing~~ system of claim 4, wherein
the allocator is further configured to:

receive allocation requests from other input data-streams of the ~~plurality of~~ multiple input data-streams,

determine a relative priority of the allocation requests from the other input data-streams and the request from the selected input data-stream, and

identify the selected input data-stream, based on the relative priority.

6. (Currently amended) The ~~multiple-input queuing~~ system of claim 4, wherein
the allocator is further configured to:

receive allocation requests from other input data-streams of the ~~plurality of~~ multiple input data-streams, and

allocate other ~~memory-elements~~unused data memory locations ~~of the plurality of memory-elements~~ for storing other data-~~items~~ elements from the other input data-streams.

7. (Currently amended) The ~~multiple-input queuing~~ system of claim 6, wherein
the allocator is configured to allocate the other ~~memory-elements~~ unused data memory locations contemporaneously with allocating the ~~memory-element~~data memory location for storing the data-~~item~~ element from the selected input data-stream.

8. (Currently amended) The ~~multiple-input queuing~~ system of claim 6, wherein
the ~~mapper that is further~~output means is configured to:

receive requests for outputs corresponding to the other input data-streams,

determine addresses-~~associated with the other memory-elements~~, based on the request for the other input data-streams, and

provide the other data-~~items~~ elements ~~from the other memory-element~~ as outputs-~~from the multiple-input queuing system~~, based on the addresses associated with the other memory-element.

3

9. (Canceled)

10. (Canceled)

11. (Canceled)

4

12. (Canceled)

13. (Currently amended)  The method of claim 1~~7~~2, ~~further including~~comprising:

allocating a plurality of ~~select memory elements of the plurality of memory elements~~unused data memory locations to a plurality of select input data-streams~~of the one or more input streams~~,

storing a received data-~~item~~ element from each of the plurality of select input data-streams to a corresponding each of the plurality of ~~select memory elements~~unused data memory locations, and

storing an identification of each of the plurality of ~~select memory elements~~previously unused data memory locations corresponding to each of the plurality of select input data-streams.

14. (Currently amended) The method of claim 1~~7~~2, wherein:

storing the identification of the ~~select memory element~~previously unused data memory location includes

placing the identification in a ~~first-in-first-out~~pointer queue that is associated with the select input data-stream, and

~~providing the received~~causing a next data-~~item~~ element to be output includes

removing the identification from the ~~first-in-first-out~~pointer queue that is associated with the select input data-stream.

5

15. (Canceled)

16. (New) A shared memory queue for receiving multiple input data streams and providing a single output data stream, comprising:

an array of data memory locations;

input means for selecting an unused data memory location and routing a data element from one of the input streams to be stored therein, wherein a routing path is provided for each of the input data streams to each of the data memory locations;

respective pointer queues corresponding to respective ones of said input data streams, wherein when a data element from a given input data stream is stored in a given data memory location, a pointer to the given data memory location is stored in a pointer queue corresponding to the given input data stream;

a readout circuit coupled to the array of data memory locations; and

output means for causing a next data element of a selected input data stream to be output, by applying to the array of data memory locations a pointer from a pointer queue corresponding to the selected input data stream.

17. (New) A method of using pointer queues, an array of data memory locations and a readout circuit to queue and de-queue data from multiple input data streams to providing a single output data stream, comprising:

selecting an unused data memory location and routing a data element from one of the input streams to be stored therein, wherein a routing path is provided for each of the input streams to each of the data memory locations;

when a data element from a given input data stream is stored in a given data memory location, storing a pointer to the given data memory location in a pointer queue corresponding to the given input data stream;

causing a next data element of a selected input data stream to be output, by applying to the array of data memory locations a pointer from a pointer queue corresponding to the selected input data stream.

6